УДК 519.712

А. А. Булатов

СЛОЖНОСТЬ ЗАДАЧ О ПОДСЧЕТЕ ЧИСЛА РЕШЕНИЙ*

Введение

Количество и многообразие комбинаторных алгоритмов, постоянно возникающих в теоретической информатике и ее приложениях, вызывают естественное стремление систематизировать и унифицировать этот массив, разработать универсальные алгоритмы, не зависящие от частных особенностей той или иной задачи. Один из возможных путей сделать это – комбинаторная задача CSP[1], а также связанная с ней задача CSP о подсчете решений, обозначаемая #CSP. В задаче CSP требуется выяснить, выполнима ли в данной модели диофантова формула[2], т. е. экзистенциальная формула первой ступени, бескванторная часть которой является конъюнкцией атомарных формул; соответственно в задаче #CSP требуется найти число выполняющих присваиваний. В другой, эквивалентной, формулировке требуется определить, существует ли гомоморфизм между двумя конечными моделями и соответственно число таких гомоморфизмов. Хотя эти задачи составляют лишь малую часть многих известных комбинаторных задач, в последнее десятилетие стало ясно, что они занимают особое место в теоретической информатике и смежных областях.

Изучение задачи #CSP является составной частью исследования задач о подсчете числа решений комбинаторных задач, инициированного Валиантом [1,2]. С другой стороны, поскольку практически все классические задачи о подсчете решений, такие, как #-Выполнимость, #-Сочетания в графе, Надежность графа, Перманент, Антицепь, #H-Раскраска графа и др., естественным образом представимы в виде частных случаев задачи #CSP, эта задача отражает в себе состояние всей области.

Исследования, связанные с задачей #CSP, можно сгруппировать в три взаимосвязанных направления. Первое из них условно назовем сложностью

*Работа выполнена при поддержке межвузовской научной программы «Университеты России» (проект № 04.01.437) и президентской программы поддержки ведущих научных школ Российской Федерации (проект НШ-2227.2003.1).

[1]От английского «Constraint Satisfaction Problem»; в немногочисленных русскоязычных работах по этой проблематике встречается также название «Обобщенная выполнимость».

[2]Такие формулы называются также *примитивно позитивными*.

ограниченных задач #CSP. Общая задача #CSP является #P-полной[3], поэтому естественными являются попытки изучения некоторых подзадач этой задачи с целью описать те из них, которые могут быть решены за полиномиальное время. Данное направление представлено работами [3–10]. Второе направление нацелено на поиск алгоритмов, большей частью приближенных и/или вероятностных, для решения частных задач #CSP. Как правило, такими задачами являются задачи из теории графов. В данном направлении следует упомянуть работы [11–15]. Наконец, заметим, что число гомоморфизмов из данной конечной модели в некоторую фиксированную модель может рассматриваться как некий числовой параметр, связанный с моделью. Таким образом, третье направление изучает вопрос о том, какие числовые параметры могут быть представлены в такой форме. Пионерские работы в данном направлении принадлежат Ловашу [16].

Данная статья представляет собой краткий обзор результатов о задаче #CSP, инспирированный недавним прогрессом в изучении сложности подзадач этой задачи. Упомянутый прогресс стал возможен благодаря открытию новой связи между сложностью задачи #CSP и универсальной алгеброй. Таким образом, мы в основном сосредоточиваемся на первом из перечисленных направлений. Однако для полноты картины мы также затронем два других направления. Бо́льшая часть приводимых результатов опубликована, два результата (следствие 3.2 и теорема 3.7) сообщаются здесь впервые.

В разделе 1 даны необходимые определения, а также приведен ряд примеров задач, выразимых в виде задачи #CSP. В разделах 2 и 3 мы даем обзор результатов о сложности подзадач задачи #CSP, полученных с помощью ограничений различных типов, а также описываем связь сложности подзадач такого типа со свойствами конечных алгебр. Наконец, в разделе 4 перечислены результаты Ловаша, а также основные результаты, связанные с приближенным решением задач #CSP.

1. Определения и примеры

1.1. Задача #CSP

Существует несколько эквивалентных определений задачи CSP о подсчете решений. Наиболее удобным для нас является определение с помощью гомоморфизмов конечных моделей. Напомним, что *сигнатурой* называется конечное множество реляционных символов $\{\varrho_i \mid i \in I\}$, каждый из которых имеет фиксированную арность. *Моделью* сигнатуры $\{\varrho_i \mid i \in I\}$ называется пара $\mathcal{H} = (H; \{\varrho_i^{\mathcal{H}} \mid i \in I\})$, где H – непустое множество, а каждое из

[3] Определение см. в п. 1.2.

$\varrho_i^{\mathcal{H}}$ является отношением на H, имеющим ту же арность, что и символ ϱ_i. Пусть $\mathcal{G}, \mathcal{H}$ – модели одинаковой сигнатуры $\{\varrho_i \mid i \in I\}$. *Гомоморфизмом* из $\mathcal{G}$ в $\mathcal{H}$ называется отображение $\varphi\colon G \to H$ такое, что для каждого отношения $\varrho^{\mathcal{G}}$ модели $\mathcal{G}$ и каждого кортежа $(\mathbf{a}[1], \dots, \mathbf{a}[m]) \in \varrho^{\mathcal{G}}$ выполняется $(\varphi(\mathbf{a}[1]), \dots, \varphi(\mathbf{a}[m])) \in \varrho^{\mathcal{H}}$.

Определение 1.1. Задача #CSP – *это задача, в которой по данной паре $\mathcal{G}, \mathcal{H}$ моделей конечной сигнатуры требуется определить, сколько существует гомоморфизмов из $\mathcal{G}$ в $\mathcal{H}$.*

Пусть $\mathfrak{G}$ и $\mathfrak{H}$ – классы конечных моделей. Через $\#\mathrm{CSP}(\mathfrak{G}, \mathfrak{H})$ *обозначается множество частных задач CSP, в которых условием является пара $\mathcal{G}, \mathcal{H}$ такая, что $\mathcal{G} \in \mathfrak{G}$, $\mathcal{H} \in \mathfrak{H}$.*

Если $\mathfrak{H}$ ($\mathfrak{G}$) – класс всех конечных моделей, то мы пишем $\mathrm{CSP}(*, \mathfrak{H})$ [*соответственно* $\#\mathrm{CSP}(\mathfrak{G}, *)$] *вместо* $\#\mathrm{CSP}(\mathfrak{G}, \mathfrak{H})$.

Если $\mathfrak{H}$ ($\mathfrak{G}$) – одноэлементный класс, состоящий из модели $\mathcal{H}$ [соответственно $\mathcal{G}$], то мы пишем $\#\mathrm{CSP}(*, \mathcal{H})$ [*соответственно* $\#\mathrm{CSP}(\mathcal{G}, *)$].

Пример 1.1 (#k-Выполнимость [1, 2, 5, 17]). Условием задачи #k-Выполнимость служит формула логики высказываний в конъюнктивной нормальной форме, каждая элементарная дизъюнкция которой содержит точно k литералов. Спрашивается, сколько существует наборов значений переменных, при которых формула истинна. Нетрудно видеть, что эта задача эквивалентна задаче $\#\mathrm{CSP}(*, \mathcal{S})$, где $\mathcal{S} = (\{0,1\}; \varrho_1, \dots, \varrho_{2^k})$, а $\varrho_1, \dots, \varrho_{2^k}$ – наборы выполняющих присваиваний для каждой из 2^k возможных элементарных дизъюнкций длины k. Хорошо известно [1, 2], что #k-Выполнимость #P-полна для любого $k \geq 2$.

Пример 1.2 (#H-Раскраска графа [8, 18, 19]). Пусть H – (ориентированный) граф. В задаче #H-Раскраска графа требуется найти число гомоморфизмов из данного графа G в H. Для (ориентированного) графа $\mathcal{H} = (V; E)$ задача #H-Раскраска графа эквивалентна задаче $\#\mathrm{CSP}(*, \mathcal{H})$, если $\mathcal{H}$ рассматривается как модель с одним бинарным отношением.

Пример 1.3 (Антицепь [20]). В задаче Антицепь дано конечное частично упорядоченное множество $\mathcal{P} = (P; \leq)$; требуется найти число антицепей в нем. Эта задача может быть представлена в виде задачи #CSP следующим образом. Пусть $R_\prec$ – естественный порядок на множестве $A = \{0, 1\}$, т. е. $R_\prec = \{(0,0), (0,1), (1,1)\}$. Тогда число антицепей в $\mathcal{P}$ равно числу гомоморфизмов из $\mathcal{P}$ в $\mathcal{A} = (A; R_\prec)$. Чтобы показать это, заметим, что для каждого гомоморфизма φ если $\varphi(a) = 1$ и $a \leq b$, то $\varphi(b) = 1$. Таким образом, множество $F_\varphi = \{a \in P \mid \varphi(a) = 1\}$ является фильтром P, а следовательно φ взаимно однозначно соответствует антицепи, состоящей из минимальных элементов F.

С другой стороны, каждая частная задача #CSP$(*, \mathcal{A})$ сводима к частной задаче Антицепь, хотя и более сложным образом: для заданной модели $\mathcal{G} = (G; R)$ с единственным бинарным отношением нужно рассмотреть рефлексивно-транзитивное замыкание R, которое является квазипорядком на G; условием соответствующей задачи Антицепь служит частично упорядоченное множество симметричных блоков этого квазипорядка.

Пример 1.4 (#Линейные уравнения [17]). В этой задаче дана система линейных уравнений над конечным полем GF(n), требуется найти число решений системы. Легко видеть, что данная задача эквивалентна задаче #CSP$(*, \mathfrak{L})$, где $\mathfrak{L}$ – класс моделей конечной сигнатуры с носителем GF(n), отношения которых представимы в виде гиперплоскостей в конечномерных линейных пространствах над GF(n).

1.2. Классы сложности и сводимости

Классы сложности для задач о подсчете решений аналогичны классам сложности задач распознавания. Так, класс FP состоит из всех задач, решаемых за полиномиальное время. Однако чтобы определить класс, соответствующий NP, нам потребуется следующее представление класса NP. Пусть Σ – конечный алфавит, Σ^* – множество всех слов над Σ. Бинарное отношение R на Σ^* называется *полиномиально сбалансированным*, если существует полином p такой, что для любого $(u, v) \in R$ справедливо $|v| \le p(|u|)$; отношение R называется *полиномиально разрешимым*, если существует детерминированная машина Тьюринга T, которая для любой пары $(u, v) \in \Sigma^* \times \Sigma^*$ за полиномиальное время решает, принадлежит ли данная пара отношению R. Класс NP состоит из всех языков L над конечным алфавитом, для которых существует полиномиально сбалансированное и полиномиально разрешимое отношение R такое, что $L = \{u \mid (u, v) \in R\}$. Каждой задаче распознавания L из класса NP можно поставить в соответствие задачу о подсчете числа решений следующим образом. Пусть R – соответствующее L полиномиально сбалансированное и полиномиально разрешимое отношение. Тогда в задаче $\#L$ для данного u требуется найти число $\#u$ элементов в $\{v \mid (u, v) \in R\}$. Очевидно, $\#L$ зависит также от выбора отношения R, однако во всех случаях, с которыми мы будем иметь дело, этот выбор является в определенном смысле наиболее естественным. Задачи вида $\#L$ для всевозможных $L \in$NP образуют класс #P.

Существуют два основных типа сводимости между задачами о подсчете числа решений. *Скупой сводимостью* задачи $\#L$ над алфавитом Σ к задаче $\#M$ над алфавитом Θ называется полиномиально вычислимая функция $f\colon \Sigma^* \to \Theta^*$ такая, что $\#u = \#f(u)$ для всех $u \in \Sigma^*$. Задача $\#L$ сводится к $\#M$ *по Тьюрингу*, если существует решающая $\#L$ машина Тьюринга, использующая $\#M$ в качестве оракула и работающая полиномиальное время (каждый вызов $\#M$ считается одним шагом). Однако ни один из этих типов не может быть признан удовлетворительным. Скупая сводимость слишком слаба и не позволяет выяснить сложность многих задач, представляющих практический интерес. Сводимость по Тьюрингу слишком сильна и позволяет сводить к задачам из #P задачи, там не содержащиеся [21]. Тем не менее сводимость по Тьюрингу является наиболее используемым типом сводимости при изучении задач о подсчете числа решений. Мы также будем использовать этот тип сводимости. Задача называется *#P-трудной*, если любая задача из $\#P$ сводится к ней. Отметим, что #P-трудными могут быть и задачи о вычислении функций, не являющиеся задачами о подсчете числа решений. Задача называется *#P-полной*, если она #P-трудна и содержится в #P.

Кроме точных детерминированных алгоритмов в задачах о подсчете решений широко используются приближенные и/или вероятностные алгоритмы. Следующие понятия часто рассматриваются в качестве одних из наилучших моделей практически реализуемых вычислений. Говорят, что задача L (о подсчете решений) решается *вполне полиномиальной приближенной схемой* (ВППС), если существует такой алгоритм A, что для любого входа x задачи L и любого неотрицательного числа ϵ выполняется неравенство

$$\frac{|\#x - A(x)|}{\#x} < \epsilon,$$

где $\#x$ обозначает точное решение задачи L для входа x, и A полиномиален относительно длины x и величины $1/\epsilon$. Говорят, что L решается *вполне полиномиальной вероятностной приближенной схемой* (ВПВПС), если существует вероятностный алгоритм A с теми же свойствами, что и для ВППС, но неравенство верно с вероятностью не меньше $1/3$.

Точное определение класса задач (не только о подсчете решений, но и, более общо, о вычислении функций), решаемых с помощью ВПВПС, дано в [22]. Нам потребуется также класс RP всех задач распознавания, решаемых за полиномиальное время с помощью вероятностного алгоритма.

Основными проблемами, которые мы обсуждаем в данной статье, являются характеризация задач CSP о подсчете числа решений, принадлежащих FP, #P-полных, а также решаемых с помощью ВПВПС. Если для первых двух проблем получены достаточно серьезные продвижения, то в поиске ВПВПС прогресс не столь значителен. Основные результаты в этой области заключаются, как правило, в нахождении таких алгоритмов для конкретных задач.

2. Задачи вида $\#\mathrm{CSP}(\mathfrak{G}, *)$

Нам потребуется несколько дополнительных определений. Пусть $\mathcal{G} = (G; R_1^{\mathcal{G}}, \ldots, R_n^{\mathcal{G}})$ – конечная модель. *Графом Гаифмана* $\mathcal{G}$ называется граф $G(\mathcal{G})$, множеством вершин которого является G, а вершины $a, b \in G$ смежны тогда и только тогда, когда a, b принадлежат одному и тому же кортежу какого-либо из отношений $R_1^{\mathcal{G}}, \ldots, R_n^{\mathcal{G}}$.

Далее, напомним, что *древесным разбиением* графа $G = (V, E)$ называется разбиение множества его вершин $\{C_w \mid w \in W\}$, индексированное вершинами некоторого дерева $T = (W, D)$ и удовлетворяющее следующим двум условиям: для любого ребра $(a, b) \in E$ найдется $w \in W$ такая, что $a, b \in C_w$; если $a \in C_v \cap C_w$, то a принадлежит C_u для всех вершин u, лежащих на пути от v к w. *Шириной* древесного разбиения называется размер наибольшего блока разбиения. Граф имеет *ширину* k, если существует его древесное разбиение ширины k.

Следующая теорема исчерпывающе характеризует задачи вида $\#\mathrm{CSP}(\mathfrak{G}, *)$, решаемые за полиномиальное время.

Теорема 2.1 (Далмау, Йонссон [6]). *Задача $\#\mathrm{CSP}(\mathfrak{G}, *)$ принадлежит FP тогда и только тогда, когда существует k такое, что граф Гаифмана любой модели из $\mathfrak{G}$ имеет древесную ширину k.*

3. Задачи вида $\#\mathrm{CSP}(*, \mathfrak{H})$

Этот раздел посвящен результатам, показывающим, что каждая задача вида $\#\mathrm{CSP}(*, \mathfrak{H})$ либо принадлежит FP, либо #P-полна, а также результатам, устанавливающим критерий полиномиальной разрешимости таких задач.

Первым результатом в этом направлении явилась характеризация двухэлементных #-полиномиальных моделей.

Теорема 3.1 (Кренью, Херрман [5, 17]). *Задача $\#\mathrm{CSP}(*, \mathcal{H})$ для двухэлементной модели $\mathcal{H}$ принадлежит FP тогда и только тогда, когда каждое отношение из $\mathcal{H}$ является множеством решений некоторой системы линейных уравнений над* $\mathrm{GF}(2)$. *В противном случае эта задача #P-полна.*

Другим значительным достижением на этом пути стала следующая теорема, описывающая неориентированные графы $\mathcal{H}$, для которых задача #H-Раскраска графа может быть решена за полиномиальное время. *Рефлексивным* мы называем граф, у которого каждая вершина имеет петлю.

Теорема 3.2 (Дайер, Гринхилл [8]). *Задача $\#H$-Раскраска для неориентированного графа H может быть решена за полиномиальное время тогда и только тогда, когда каждая компонента связности H есть либо изолированная вершина, либо полный двудольный граф без петель, либо полный рефлексивный граф. В противном случае эта задача #P-полна.*

К сожалению, методы, использованные Кренью и Херрманом, а также Дайером и Гринхилл, годятся лишь в случае двухэлементных моделей и неориентированных графов. В общем случае нужна более мощная техника, которая описана в следующем разделе.

3.1. Задача #CSP и конечные алгебры

Прежде всего мы покажем, что каждой конечной модели $\mathcal{H}$ может быть сопоставлена конечная алгебра таким образом, что сложность задачи $\#\mathrm{CSP}(*,\mathcal{H})$ полностью определяется свойствами этой алгебры. Напомним, что операция $f(x_1,\dots,x_n)$ *сохраняет* отношение ϱ (говорят также, что ϱ *инвариантно* относительно f, а f есть *полиморфизм* ϱ), если для любых $\mathbf{a}_1,\dots,\mathbf{a}_n\in\varrho$, кортеж $f(\mathbf{a}_1,\dots,\mathbf{a}_n)$, где f действует покомпонентно, также содержится в ϱ. Для множества операций C и конечной модели $\mathcal{H}=(H;\{\varrho_i \mid i\in I\})$ множество всех отношений, инвариантных относительно операций из C, обозначается через $\mathsf{Inv}(C)$, а множество всех операций, сохраняющих все отношения из $\{\varrho_i \mid i\in I\}$, – через $\mathsf{Pol}(\mathcal{H})$. Наконец, через $\mathbb{A}_{\mathcal{H}}$ обозначается алгебра $(H;\mathsf{Pol}(\mathcal{H}))$.

Теорема 3.3 (Булатов, Далмау [3]). *Пусть $\mathcal{G},\mathcal{H}$ – конечные модели, имеющие один и тот же носитель, и такие, что $\mathbb{A}_{\mathcal{G}}$ является редуктом $\mathbb{A}_{\mathcal{H}}$. Тогда $\#\mathrm{CSP}(*,\mathcal{H})$ сводима по Тьюрингу к $\#\mathrm{CSP}(*,\mathcal{G})$.*

Теорема 3.3 позволяет сформулировать понятие #-полиномиальной алгебры. Алгебра $\mathbb{A}$ называется *#-полиномиальной*, если для каждой конечной модели $\mathcal{H}$ такой, что $\mathbb{A}$ является редуктом $\mathbb{A}_{\mathcal{H}}$, задача $\#\mathrm{CSP}(*,\mathcal{H})$ может быть решена за полиномиальное время. Алгебра $\mathbb{A}$ называется *#P-полной*, если существует конечная модель $\mathcal{H}$ такая, что $\#\mathrm{CSP}(*,\mathcal{H})$ #P-полна и $\mathbb{A}$ является редуктом $\mathbb{A}_{\mathcal{H}}$.

Свойство #-полиномиальности выдерживает многие стандартные алгебраические манипуляции.

Предложение 3.1 (Булатов, Далмау [3]). *Пусть $\mathbb{A}$ – конечная алгебра, $\mathbb{B}$, $\mathbb{C}$, $\mathbb{A}^n$ – ее подалгебра, гомоморфный образ и n-я прямая степень соответственно. Если алгебра $\mathbb{A}$ является #-полиномиальной, то алгебры $\mathbb{B}$, $\mathbb{C}$, $\mathbb{A}^n$ также #-полиномиальны. Если $\mathbb{B}$, $\mathbb{C}$ или $\mathbb{A}^n$ являются #P-полными, то алгебра $\mathbb{A}$ также #P-полна.*

Предложение 3.1 влечет, что #-полиномиальность – это свойство многообразий алгебр.

Следствие 3.1. *Пусть* $\mathfrak{A}$ *– многообразие, порожденное конечной алгеброй* $\mathbb{A}$. *Если* $\mathbb{A}$ *является #-полиномиальной, то любая конечная алгебра из* $\mathfrak{A}$ *также #-полиномиальна. Если* $\mathfrak{A}$ *содержит конечную #P-полную алгебру, то сама алгебра* $\mathbb{A}$ *является #P-полной.*

Напомним, что *полным идемпотентным редуктом* алгебры $\mathbb{A} = (A; C)$ называется алгебра $\mathsf{Id}(\mathbb{A}) = (A; C_{\mathrm{id}})$, где C_{id} обозначает множество идемпотентных термальных операций $\mathbb{A}$.

Предложение 3.2 (Булатов, Далмау [3]). *Конечная алгебра* $\mathbb{A}$ *является #-полиномиальной [#P-полной] тогда и только тогда, когда* $\mathsf{Id}(\mathbb{A})$ *является #-полиномиальной [#P-полной].*

3.2. Критерий #-полиномиальности

Из следствия 3.1 вытекает, что #-полиномиальность алгебры должна определяться некоторым набором тождеств. Одно из таких тождеств давно известно. Тернарная операция f называется *мальцевской*, если она удовлетворяет тождествам

$$f(x, y, y) = f(y, y, x) = x.$$

Отметим, что наличие мальцевского терма в алгебре эквивалентно конгруэнц-перестановочности многообразия, ею порождаемого.

Первое необходимое условие #-полиномиальности мы приведем, используя следующую задачу #CSP. Пусть ϱ – бинарное рефлексивное, но не симметричное отношение на конечном множестве H, содержащем не менее двух элементов; пусть также $\mathcal{H} = (H; \varrho)$ – соответствующая модель. Нетрудно видеть, что задача #CSP$(*, \mathcal{H})$ является обобщением задачи Антицепь из примера 1.3.

Предложение 3.3 (Булатов, Далмау [3]). *Задача* #CSP$(*, \mathcal{H})$ *является #P-полной.*

Теперь мы воспользуемся следующим результатом.

Теорема 3.4 (Хагеман, Мичке [23]). *Для многообразия* $\mathfrak{A}$ *и числа* n *следующие условия эквивалентны.*

(1) *Существуют термы* $p_1(x, y, z), \ldots, p_n(x, y, z)$ *такие, что* $\mathfrak{A}$ *удовлетворяет тождествам (a)* $x = p_1(x, y, y)$, *(b)* $p_i(x, x, y) = p_{i+1}(x, y, y)$ *для каждого* i, *(c)* $p_n(x, x, y) = y$.

(2) *Для любой алгебры* $\mathbb{A} = (A; F) \in \mathfrak{A}$ *и произвольного рефлексивного отношения* $R \in \mathsf{Inv}F$ *выполняется условие* $R^{-1} \subseteq R^n$.

Заметим, что при $n = 1$ отношения из третьего условия теоремы 3.4 – это в точности рефлексивные, но не симметричные отношения; а (единственный) терм из второго условия – это мальцевская операция. Таким образом, из теоремы 3.4 и предложения 3.3 вытекает необходимое условие #-полиномиальности алгебры.

Теорема 3.5. *Каждая конечная #-полиномиальная алгебра является мальцевской.*

Однако, как показывает следующий пример, это условие не является достаточным.

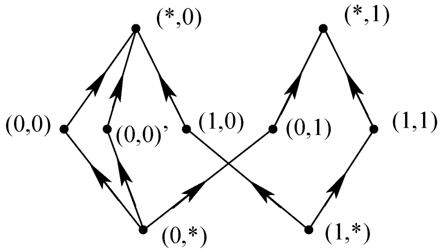

Рис. 1

Пример 3.1. Рассмотрим задачу $\#H$-Раскраска графа, где H – граф, изображенный на рис. 1. Заметим, что вершины H расположены на трех уровнях, которые мы будем называть нижним, средним и верхним. Определим тернарную операцию, сохраняющую H, в два шага. Сначала мы отождествим вершины $(0,0)$ и $(0,0)'$, сохраняя для «совмещенной» вершины обозначение $(0,0)$. Определим тернарную операцию g на полученном 8-элементном множестве следующим образом. Для любых $x_1 = (i_1, j_1)$, $x_2 = (i_2, j_2)$, $x_3 = (i_3, j_3)$, где $i_1, i_2, i_3, j_1, j_2, j_3 \in \{*, 0, 1\}$, полагаем $k_1 = i_1 - i_2 + i_3 \pmod 2$ (если $i_1, i_2, i_3 \in \{0, 1\}$), $k_2 = j_1 - j_2 + j_3 \pmod 2$ (если $j_1, j_2, j_3 \in \{0, 1\}$), и

$$g(x_1, x_2, x_3) = \begin{cases} (k_1, k_2), & \text{если } x_1, x_2, x_3 \text{ находятся на среднем уровне;} \\ x_\ell, & \text{если в точности два из } x_1, x_2, x_3 \text{ находятся на} \\ & \text{одном и том же уровне, а } x_\ell \text{ на другом;} \\ (*, k_2), & \text{если } x_1, x_2, x_3 \text{ находятся на верхнем уровне;} \\ (k_1, *), & \text{если } x_1, x_2, x_3 \text{ находятся на нижнем уровне;} \\ x_1 & \text{в противном случае.} \end{cases}$$

На втором шаге определим операцию на H. Для этого положим $m(x_1, x_2, x_3) = g(x_1, x_2, x_3)$, если $g(x_1, x_2, x_3) \neq (0,0)$ (таким образом, мы игнорируем разницу между $(0,0)$ и $(0,0)'$ в этом случае); в противном случае полагаем $m(x_1, x_2, x_3) = (0,0)'$, если $x_1 = x_2 = x_3 = (0,0)'$ или в точности один из x_1, x_2, x_3 равен $(0,0)'$, а два других равны между собой; в остальных случаях полагаем $m(x_1, x_2, x_3) = (0,0)$.

Нетрудно проверить, что полученная операция является мальцевским полиморфизмом графа H. Однако, как показано в [4], #P-полная задача #Максимальный разрез сводится к #H-Раскраске; следовательно, эта задача также #P-полна.

Второе необходимое условие #-полиномиальности носит комбинаторный характер. Пусть α, β – отношения эквивалентности на некотором множестве. Пусть также $A_1, \ldots, A_k$, $B_1, \ldots, B_\ell$ суть α- и β-классы соответственно. Через $M(\alpha, \beta)$ мы обозначаем матрицу (m_{ij}) размеров $k \times \ell$, где $m_{ij} = |A_i \cap B_j|$.

Теорема 3.6 (Булатов, Гроэ [4]). *Пусть α, β – отношения эквивалентности на некотором множестве. Задача $\#\mathrm{CSP}(\{\alpha, \beta\})$ #-полиномиальна тогда и только тогда, когда α, β перестановочны и $\mathsf{rank}(M(\alpha, \beta))$ равен количеству классов в $\alpha \vee \beta$. В противном случае $\#\mathrm{CSP}(\{\alpha, \beta\})$ #P-полна.*

Пусть теперь $\mathbb{A}$ – мальцевская алгебра. Для любых ее конгруэнций α, β имеем $\#\mathrm{CSP}(\{\alpha, \beta\}) \subseteq \#\mathrm{CSP}(\mathbb{A})$. Таким образом, если $\mathsf{rank}(M(\alpha, \beta))$ не равен количеству классов в $\alpha \vee \beta$ (поскольку $\mathbb{A}$ мальцевская, α, β перестановочны), то $\#\mathrm{CSP}(\mathbb{A})$ #P-полна. Этот факт оправдывает следующее определение. Алгебра называется *конгруэнц-сингулярной*, если для любых ее двух конгруэнций α, β $\mathsf{rank}(M(\alpha, \beta))$ равен количеству классов в $\alpha \vee \beta$. Далее, многообразие называется *конгруэнц-сингулярным*, если каждая его конечная алгебра конгруэнц-сингулярна.

Комбинируя теорему 3.6 и предложение 3.1, мы получаем следующее

Следствие 3.2. *Если конечная алгебра #-полиномиальна, то она порождает конгруэнц-сингулярное многообразие.*

Наконец, может быть показано, что два сформулированных необходимых условия #-полиномиальности являются также и достаточными. Мы получаем, таким образом, критерий #-полиномиальности конечных алгебр.

Теорема 3.7. *Конечная алгебра является #-полиномиальной тогда и только тогда, когда она порождает конгруэнц-перестановочное и конгруэнц-сингулярное многообразие. В противном случае $\mathbb{A}$ #P-полна.*

Мы завершаем этот раздел демонстрацией общей идеи алгоритма, решающего задачу $\#\mathrm{CSP}(*, \mathcal{H})$ в тех случаях, когда это можно сделать за полиномиальное время. Мы рассмотрим случай, когда $\mathcal{H} = (H; \alpha, \beta)$ – модель, отношениями которой являются два отношения эквивалентности, причем отношение $\alpha \vee \beta$ полное.

Пусть $M(\alpha, \beta) = (m_{ij})$ и $\mathsf{rank}(M(\alpha, \beta)) = 1$; пусть также $A_1, \dots, A_n$ и $B_1, \dots, B_m$ – классы α и β соответственно. Тогда существуют числа $a_1, \dots, a_n$ (например, $a_i = m_{i1}$) и $b_1, \dots, b_m$ (например, $b_j = \frac{m_{1j}}{m_{11}}$) такие, что $m_{ij} = a_i \cdot b_j$. Возьмем модель $\mathcal{G} = (G; \alpha', \beta')$ и обозначим через α^*, β^* наименьшие отношения эквивалентности на G, содержащие α', β' соответственно. Классы эквивалентностей α^*, β^* мы будем обозначать через $C_1, \dots, C_s$ и $D_1, \dots, D_t$ соответственно. Легко понять, что отображение $\varphi\colon G \to H$ является гомоморфизмом из $\mathcal{G}$ в $\mathcal{H}$ тогда и только тогда, когда φ отображает каждый класс α^* в класс α и каждый класс β^* в класс β. Следовательно, φ соответствует паре отображений $\varphi_1\colon \{1, \dots, s\} \to \{1, \dots, n\}$ и $\varphi_2\colon \{1, \dots, t\} \to \{1, \dots, m\}$. Таким образом, общее число гомоморфизмов равно

$$
\begin{aligned}
&\sum_{\substack{\varphi_1\colon \{1,\dots,s\}\to\{1,\dots,n\}\\ \varphi_2\colon \{1,\dots,t\}\to\{1,\dots,m\}}} \prod_{\substack{i\in\{1,\dots,s\}\\ j\in\{1,\dots,t\}}} m_{\varphi_1(i)\varphi_2(j)}^{|C_i\cap D_j|} = \\
&= \sum_{\substack{\varphi_1\colon \{1,\dots,s\}\to\{1,\dots,n\}\\ \varphi_2\colon \{1,\dots,t\}\to\{1,\dots,m\}}} \prod_{\substack{i\in\{1,\dots,s\}\\ j\in\{1,\dots,t\}}} a_{\varphi_1(i)}^{|C_i\cap D_j|} b_{\varphi_2(j)}^{|C_i\cap D_j|} = \\
&= \sum_{k=1}^{n} a_k^{|C_1|} \cdot \sum_{\substack{\varphi_1'\colon \{2,\dots,s\}\to\{1,\dots,n\}\\ \varphi_2\colon \{1,\dots,t\}\to\{1,\dots,m\}}} \prod_{\substack{i\in\{2,\dots,s\}\\ j\in\{1,\dots,t\}}} a_{\varphi_1(i)}^{|C_i\cap D_j|} b_{\varphi_2(j)}^{|C_i\cap D_j|} = \dots \\
&= \prod_{i=1}^{s} (a_1^{|C_i|} + \dots + a_n^{|C_i|}) \cdot \prod_{j=1}^{t} (b_1^{|D_j|} + \dots + b_m^{|D_j|}).
\end{aligned}
$$

Последнее число может быть вычислено за полиномиальное время.

3.3. Приложения

В этом разделе мы покажем, как теоремы 3.1 и 3.2 могут быть получены с помощью теоремы 3.7.

Начнем с теоремы 3.1. Если задача $\#\mathrm{CSP}(*, \mathcal{H})$ для двухэлементной модели $\mathcal{H}$ может быть решена за полиномиальное время, то по теореме 3.7 она имеет мальцевский полиморфизм. Далее, из результатов Поста [24] вытекает, что если двухэлементная модель имеет мальцевский полиморфизм, то она имеет и полиморфизм вида $x - y + z$, где $+, -$ суть операции двухэлементного поля. Таким образом, мы получили необходимое условие теоремы 3.1. Чтобы убедиться в достаточности этого условия, отметим, что любое отношение, инвариантное относительно $x-y+z$, является множеством решений некоторой системы линейных уравнений (см. пример 1.4).

В случае теоремы 3.2 трудной частью также является доказательство #P-полноты. Возьмем связный граф $\mathcal{H} = (H, E)$ такой, что задача #H-Раскраска (т. е. #CSP$(*, \mathcal{H})$) решается за полиномиальное время. По теореме 3.7 он имеет мальцевский полиморфизм d. Отметим сначала, что этот граф не содержит вершин a, b, c, d (не обязательно различных) таких, что $(a, c), (a, d), (b, d)$ – ребра, а (b, c) – нет. В самом деле, если это не так, то

$$d\left(\begin{pmatrix} a \\ c \end{pmatrix}, \begin{pmatrix} a \\ d \end{pmatrix}, \begin{pmatrix} b \\ d \end{pmatrix}\right) = \begin{pmatrix} b \\ c \end{pmatrix} \notin E,$$

противоречие с тем, что d – полиморфизм. Мы рассмотрим два случая.

Случай 1. $\mathcal{H}$ имеет петлю.

Предположим, что $\mathcal{H}$ имеет петлю в вершине u и v – смежная с u вершина. Тогда, выбирая $a = u$, $b = v$, $c = v$ и $d = u$, получаем, что $\mathcal{H}$ содержит петлю (v, v). Поскольку $\mathcal{H}$ связен, все его вершины имеют петлю. Далее, пусть $(u, v), (v, w)$ – путь длины 2. Мы полагаем $a = v$, $b = w$, $c = u$ и $d = v$ и получаем, что $\mathcal{H}$ содержит ребро (u, w). Таким образом, $\mathcal{H}$ – полный рефлексивный граф.

Случай 2. $\mathcal{H}$ не имеет петель.

Пусть $(t, u), (u, v), (v, w)$ – путь длины 3. Выберем $a = t$, $b = u$, $c = v$ и $d = w$. Таким образом, (t, w) является ребром графа. Следовательно, если $\mathcal{H}$ двудольный, то он полный двудольный граф, а если он не двудольный, то содержит треугольник. Покажем, что последний случай невозможен. Если u, v, w составляют треугольник, то, полагая $a = u$, $b = c = v$, $d = w$, мы заключаем, что $\mathcal{H}$ содержит петлю (v, v), что противоречиво.

4. Приближенные алгоритмы и числовые параметры моделей

4.1. Задачи, решаемые приближенными алгоритмами

К сожалению, весьма немногие задачи о подсчете решений попадают в класс FP. Кроме полиномиально разрешимых задач, описанных в разд. 2 и 3, известны лишь считанные примеры таких задач (их представление в виде задач #CSP менее очевидно). Это задача #Остов графа о подсчете числа остовных деревьев, полиномиальная разрешимость которой следует из теоремы Кирхгофа, и задача #Паросочетания планарного графа [25]. Валиант заметил, что обе эти задачи сводятся к решению систем линейных уравнений. Поэтому использование приближенных и вероятностных алгоритмов является здесь весьма привлекательным.

Хорошо известно, что для задач оптимизации существует огромное число различных классов сложности, объединяющих задачи, решаемые с различной степенью приближения. Любопытно, что для задач о подсчете решений это, по всей видимости, не так. Одно из свидетельств в пользу этого – результат Джеррума и Синклера из [26]. Полиномиально сбалансированное полиномиально разрешимое отношение ϱ называется *самосводимым*, если множество $\{y \mid (x,y) \in \varrho\}$ может быть представлено определенным образом с помощью множеств $\{y \mid (x_1,y) \in \varrho\}, \dots, \{y \mid (x_k,y) \in \varrho\}$, где $|x_i| < |x|$ для $1 \le i \le k$.

Теорема 4.1 (Джеррум, Синклер [26]). *Пусть L – задача о подсчете решений, соответствующая полиномиально сбалансированному и полиномиально разрешимому самосводимому отношению. Пусть также E – это множество всех $\alpha \in \mathbb{R}$, для которых существует $\delta > 0$ и полиномиальный вероятностный алгоритм A такие, что A с вероятностью $\frac{1}{2} + \delta$ вычисляет $\#x$ с относительной погрешностью, не превосходящей $|x|^\alpha$. Тогда E либо пусто, либо равно $\mathbb{R}$.*

Таким образом, наибольший интерес для приближенного решения задач о подсчете решений представляют ВПВПС. Несколько таких алгоритмов хорошо известны. Назовем алгоритм Лубы и Великовича [27] для решения задачи #Двойственная выполнимость, в которой требуется найти число выполняющих присваиваний для пропозициональной формулы в ДНФ; алгоритм Карзанова и Хачияна [28] для вычисления числа линейных расширений частичного порядка; алгоритм Джеррума и Синклера [29] для вычисления перманента плотной матрицы, элементы которой содержатся в множестве $\{0,1\}$; а также алгоритм Джеррума [30] для задачи #k-Раскраска графа, ограниченной на множество графов, степени которых ограничены.

Во всех указанных случаях ВПВПС могут быть получены из алгоритмов, выбирающих комбинаторные объекты (например, паросочетания) из большого массива (например, из множества всех паросочетаний графа) в соответствии с некоторым вероятностным распределением (например, равномерным). Алгоритмы данного вида обычно реализуются с помощью марковских цепей, состояниями которых являются нужные нам объекты, обладающих следующим свойством: вне зависимости от начального объекта вероятность того, что через фиксированное число шагов цепь окажется в некотором состоянии, соответствует заданному распределению. В [26, 29] были найдены условия, при которых марковские цепи алгоритмов поиска выборки могут быть трансформированы в ВПВПС для подсчета числа решений.

Для некоторых задач существование ВПВПС эквивалентно равенству RP=NP. Таковыми, например, являются задача #Выполнимость [31], задача #Независимое множество [13], в которой требуется найти число независимых множеств графа (указанная эквивалентность остается верной, даже если эта задача ограничена на класс графов, степени вершин которых не превосходят 25). В работе [13] было также указано несколько задач, в их числе Антицепь и #Двудольное независимое множество (о числе независимых множеств в двудольных графах), для которых неизвестны ВПВПС, но и крайне маловероятно, что они столь же трудны, как и задачи #Выполнимость и #Независимое множество.

4.2. Числовые параметры моделей

Такие числа, связанные с графами, как количество клик, независимых множеств, разрезов и т. п., являются важными характеристиками графа, а аналогичные им свойства – важными характеристиками произвольных конечных моделей. Все они могут быть вычислены с помощью задач CSP о подсчете числа решений. Возникает естественный вопрос: какие числовые характеристики графов (или, более общо, конечных моделей) могут быть представлены с помощью задач CSP о подсчете числа решений? Частичному решению этой проблемы посвящена недавняя работа Ловаша с соавторами [16]. Заметим, что в этой работе рассматривается задача, несколько более общая, чем задача о подсчете гомоморфизмов графов, а именно задача о нахождении суммарного веса гомоморфизмов реберно взвешенных графов. Однако, как показано в [4], задача о взвешенных графах может быть сведена к обычной задаче #CSP.

Нам потребуется еще несколько определений. Для графов $\mathcal{G}_1$ и $\mathcal{G}_2$ через $\mathcal{G}_1 \cup \mathcal{G}_2$ обозначается их дизъюнктное объединение. Граф $\mathcal{G}$ называется *k-помеченным*, если какие-либо k из его вершин помечены метками из множества $\{1, \ldots, k\}$. Изоморфизм помеченных графов – это обычный изоморфизм графов, сохраняющий метки. *Произведение* $\mathcal{G}_1\mathcal{G}_2$ двух k-помеченных графов $\mathcal{G}_1$ и $\mathcal{G}_2$ определяется следующим образом: в $\mathcal{G}_1 \cup \mathcal{G}_2$ мы отождествляем вершины, помеченные одинаково. Пусть теперь f – числовой параметр графа и $k \geq 0$. Определим (бесконечную) матрицу $P(f, k)$ следующим образом. Строки и столбцы этой матрицы помечены изоморфными классами k-помеченных графов, а число, находящееся в столбце, помеченном $\mathcal{G}_1$, и строке, помеченной $\mathcal{G}_2$, равно $f(\mathcal{G}_1\mathcal{G}_2)$. В общем случае ранг $P(f, k)$ бесконечен, однако для большинства важных параметров он конечен. Мы будем рассматривать лишь этот случай. Тогда мы можем определить функцию $r_f(k) = \mathsf{rank}(P(f, k))$. В этом случае мы можем считать, что $P(f, k)$ – конечная матрица.

Теорема 4.2 (Ловаш и др. [16]). *Параметр f может быть представлен как решение задачи $\#\mathrm{CSP}(*, \mathcal{H})$ для некоторого конечного графа $\mathcal{H}$ тогда и только тогда, когда функция r_f ограничена некоторой экспоненциальной функцией, а матрица $P(f, k)$ позитивна и все ее собственные значения неотрицательны.*

Литература

1. Valiant L. The complexity of computing the permanent // Theoretical Computing Science. 1979. Vol. 8. P. 189–201.
2. Valiant L. The complexity of enumeration and reliability problems // SIAM J. Computing. 1979. Vol. 8, № 3. P. 410–421.
3. Bulatov A. A., Dalmau V. Towards a dichotomy theorem for the counting constraint satisfaction problem // The 44th IEEE Symposium on Foundations of Computer Science, FOCS'03. IEEE Computer Society, 2003. P. 562–571.
4. Bulatov A. A., Grohe M. The complexity of partition functions // The 31st International Colloquium on Automata, Languages and Programming, ICALP'04. 2004. P. 294–306.
5. Creignou N., Hermann M. Complexity of generalized satisfiability counting problems // Information and Computation. 1996. Vol. 125, № 1. P. 1–12.
6. Dalmau V., Jonsson P. The complexity of counting homomorphisms seen from the other side // Theoretical Computer Science. 2004. Vol. 329, № 1–3. P. 315–323.
7. Diaz J., Serna M., Thilikos D. M. Counting h-colorings of partial k-trees // Ibid. 2002. Vol. 281. P. 291–309.
8. Dyer M., Greenhill C. The complexity of counting graph homomorphisms // Random Structures and Algorithms. 2000. Vol. 17. P. 260–289.
9. Hell P. Algorithmic aspects of graph homomorphisms // Survey in Combinatorics 2003, London Math. Society Lecture Note Series. Cambridge: Cambridge University Press. Vol. 307. P. 239–276.
10. Hell P., Nešetřil J. Counting list homomorphisms for graphs with bounded degrees // Graphs, Morphisms and Statistical Physics. DIMACS Series in Discrete Mathematics and Theoretical Computer Science. 2004. Vol. 63. P. 105–112.
11. Bubley R., Dyer M., Greenhill C., Jerrum M. On approximately counting colourings of small degree graphs // SIAM J. Computing. 1999. Vol. 29. P. 387–400.
12. Dyer M., Frieze A., Jerrum M. On counting independent sets in sparse graphs // SIAM J. Computing. 2002. Vol. 31. P. 1527–1541.
13. Dyer M., Goldberg L. A., Greenhill C., Jerrum M. On the relative complexity of approximate counting problems // The 3rd International Workshop on Approximation Algorithms for Combinatorial Optimization, APPROX'00. Springer-Verlag, 2000. LNCS Vol. 1913. P. 108–119.

14. Greenhill C. The complexity of counting colourings and independent sets in sparse graphs and hypergraphs // Computational Complexity. 2000. Vol. 9. P. 52–73.
15. Jerrum M., Sinclair A. The Markov chain Monte Carlo method: an approach to approximate counting and integration // Approximation Algorithms for NP-hard Problems. PSW, 1996. P. 482–520.
16. Freedman M., Lovász L., Schrijver A. Reflection positivity, rank connectivity, and homomorphism of graphs // Manuscript. 2005.
17. Creignou N., Khanna S., Sudan M. Complexity Classifications of Boolean Constraint Satisfaction Problems, volume 7 of SIAM Monographs on Discrete Mathematics and Applications. SIAM, 2001.
18. Hell P., Nešetřil J. On the complexity of H-coloring // J. Combinatorial Theory. Ser. B. 1990. Vol. 48. P. 92–110.
19. Левин Л. А. Универсальные задачи перебора // Проблемы передачи информации. 1973. Т. 9, № 3. С. 115–116.
20. Provan J. S., Ball M. O. The complexity of counting cuts and of computing the probability that a graph is connected // SIAM J. Computing. 1983. Vol. 12, № 4. P. 777–788.
21. Todá S., Ogiwara M. Counting classes are at least as hard as the polynomial-time hierarchy // Ibid. 1992. Vol. 21, № 2. P. 316–328.
22. Cook S. A., Kabanets V., Rackoff C. Efficiently approximable real-valued functions // Electronic Colloqium on Computational Complexity. 2000. Vol. 7, № 34.
23. Hagemann J., Mitschke A. On n-permutable congruences // Algebra Universalis. 1973. Vol. 3. P. 8–12.
24. Post E. L. The two-valued iterative systems of mathematical logic. Annals Mathematical Studies. Vol. 5. Princeton University Press, 1941.
25. Kasteleyn P. W. Dimer statistics and phase transitions // J. Mathematical Phisics. 1963. Vol. 4. P. 287–293.
26. Jerrum M., Sinclair A. Approximate counting, uniform generation and rapidly mixing markov chains // Information and Computation. 1989. Vol. 82. P. 93–133.
27. Luby M., Velickovic B. On deterministic approximation of DNF // Algorithmica. 1996. Vol. 16, № 4–5. P. 415–433.
28. Karzanov A., Khachiyan L. On the conductance of order Markov chains // Technical Report DCS 268, Rutgers University, 1990.
29. Jerrum M., Sinclair A. Approximating the permanent // SIAM J. Computing, 1989. Vol. 18. P. 1149–1178.
30. Jerrum M. A very simple algorithm for estimating the number of k-colourings of a low-degree graph // Technical Report ECS-LFCS-94-290, Department of Computer Science, University of Edinburgh, 1994.
31. Zuckerman L. On unapproximable versions of NP-complete problems // SIAM J. Computing. 1996. Vol. 25. P. 1293–1304.

Статья поступила 05.05.2005 г.